Formalization of Basic Combinatorics on Words

Štěpán Holub
Department of Algebra, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Štěpán Starosta
Dept. of Applied Math., Faculty of Information Technology, Czech Technical University in Prague, Czech Republic

Abstract

Combinatorics on Words is a rather young domain encompassing the study of words and formal languages. An archetypal example of a task in Combinatorics on Words is to solve the equation $x \cdot y = y \cdot x$, i.e., to describe words that commute.

This contribution contains formalization of three important classical results in Isabelle/HOL. Namely i) the Periodicity Lemma (a.k.a. the theorem of Fine and Wilf), including a construction of a word proving its optimality; ii) the solution of the equation $x^a \cdot y^b = z^c$ with $2 \leq a, b, c$, known as the Lyndon-Schützenberger Equation; and iii) the Graph Lemma, which yields a generic upper bound on the rank of a solution of a system of equations.

The formalization of those results is based on an evolving toolkit of several hundred auxiliary results which provide for smooth reasoning within more complex tasks.

2012 ACM Subject Classification Mathematics of computing → Combinatorics on words

Keywords and phrases combinatorics on words, formalization, Isabelle/HOL

Digital Object Identifier 10.4230/LIPIcs.ITP.2021.22

Supplementary Material *Software (Code Repository)*: `https://gitlab.com/formalcow/combinatorics-on-words-formalized`
archived at `swh:1:dir:78c9955742137e63eb137885a27acfd231a576f5`

Funding The authors acknowledge support by the Czech Science Foundation grant GAČR 20-20621S.

Acknowledgements We would like to thank Manuel Eberl for useful suggestions concerning formalization. We are also grateful to anonymous referees whose criticism helped to improve the presentation significantly.

1 Introduction

Combinatorics on Words usually dates its beginning (cf. Berstel and Perrin [5]) back to the works of Axel Thue on repetitions in infinite words published more than hundred years ago [34, 35]. Nevertheless, the first (collective) monograph on the subject was published only in 1983 [26]. In this paper, we are interested in the part of the field dealing with finite (rather than infinite) words, which in particular includes solving word equations (without constants). Solving general word equations is a difficult algorithmic task. Once believed to be undecidable, the first algorithm was described by Makanin in 1977 [28] (see [7] for a self-contained exposition by Diekert). Currently, the approach of *recompression* introduced by Jeż [22] is the most efficient one, with nondeterministic linear space complexity (see Jeż [23]). While the problem is NP hard, it remains a challenging open question whether it is NP complete.

We believe that combinatorics of (finite) words is an area where computer assisted formalization may be very helpful. Proofs of even fairly simple results tend to be tedious and repetitive, featuring complicated analysis of cases, which makes them hard (both for

referees and readers) to verify. Moreover, despite the short history of the field, basic auxiliary results are sometimes forgotten and rediscovered, or simply repeatedly proven in many papers. Some easily stated problems, like the solution of equations in three unknowns by Nowotka and Saarela [29, 30], or the characterization of binary equality languages by the first author [19], are nontrivial classification tasks, for which computer formalization can be decisive. Prominent examples are classification of finite groups formalized by Gonthier et al. in Coq [9], four-colour problem formalized by Gontihier also in Coq [11] or Kepler's conjecture formalized by Hales et al. in HOL Light and Isabelle [12]. Our long term ambition is to create a library of formalized results with three objectives: 1) verified basic facts (the "folklore") that can become a standard starting point for further formalization; 2) verified classical results, making sure that occasional gaps in the published proofs are not fatal, and sometimes providing polished, more straightforward proofs; 3) allowing to push boundaries of the current research in areas where a sheer complexity of the topic may be the most important barrier for further advances. Automation of repeated steps can make a crucial difference here. (In particular, we have in mind the above mentioned classification tasks.)

In this paper, we present advances in the first two of those objectives. Namely, we formalize three important classical results, which together reveal the main features of the general project of formalization of word equations. We want our formalization to reflect as clearly as possible the main ideas that would be given in (a good) paper proof. This requires an auxiliary background theory that collects humanly trivial facts about words that are nevertheless not covered by the main Isabelle/HOL library. Our auxiliary theory contains several hundred claims which we deem of fundamental nature in order to formalize some advanced results in Combinatorics on Words (see more in Section 2.4.1).

The first classical result presented in this paper is the Periodicity Lemma, also known as the theorem of Fine and Wilf [10], which regulates the possibility of a word having more than one period. It states that if a word of length at least $p + q - \gcd(p, q)$ has periods p and q, then it has also a period $\gcd(p, q)$. We present here a particularly simple proof at which we arrived through the formalization process. We take the opportunity of this simple example to illustrate some common features of our project. We have also formalized an explicit verified construction of a word witnessing that the bound given in the Periodicity Lemma is sharp. For example, the word `0102010` of length seven has periods 4 and 6 but not the period $\gcd(4, 6) = 2$, while any word of length at least eight having periods four and six has also a period two.

The second theorem deals with the equation $x^a y^b = z^c$ with $2 \leq a, b, c$. We formalize a proof that this equation admits only solutions where all unknown words x, y, and z are powers of a common word. Such solutions are called *periodic*. This classical result was first proven by Lyndon and Schützenberger [27] in a more general setting of free groups. Historically, it was the first challenging equation with three unknowns whose solutions were completely characterized. The presented proofs of the Periodicity Lemma and the solution of the Lyndon and Schützenberger equation (LSE) are mainly combinatorial.

The need to deal with equations like LSE in an *ad hoc* manner is tightly related to the fact that word equations are rather immune against the so called *defect effect*. To understand what this means, consider systems of linear equations. Each new independent linear equation decreases the degree of freedom of a solution of the corresponding system, so that n independent equations over n unknowns admit only one solution. In contrast, there is no known upper bound on the size of an independent system of word equations over $n \geq 4$ unknowns, and only a rough bound for $n = 3$ (see e.g. Saarela [31] for a survey).

The best general form of the defect effect for word equations is provided by the Graph Lemma, which is the third important result presented and formalized in this paper. We shall

discuss the Graph Lemma in detail in Section 2.3. Here, let us illustrate the main idea by an example. Consider the following system of two equations over three unknowns:

$$xyz = yzx,$$
$$xzy = zyx.$$

We construct an undirected graph whose vertices are the unknowns x, y, z. The edges, one for each equation of the system, connect first letters of left and right hand side of the equation. In our example, the edges are (x, y) and (x, z). By the Graph Lemma, such a system has periodic solutions only, since the resulting graph is connected. In other terms, since the graph has *one* connected component, all three words in any solution are powers of *one* common word. Consider, on the other hand, the system

$$xyz = zyx,$$
$$xyyz = zyyx.$$

The graph of this system has the unique edge (x, z), hence the Graph Lemma does not tell us whether the system has a non-periodic solution or not. In fact, this system has an obvious non-periodic solution $x \mapsto a$, $y \mapsto b$, $z \mapsto a$.

Our approach to the proof of the Graph Lemma exploits the algebraic concept of the *free hull* of a solution, and of its rank, that is, of the cardinality of its basis. This also means that auxiliary claims needed in the proof of the Graph Lemma are of a more algebraic flavor, compared to the proof of the Periodicity Lemma and the solution of the LSE. These claims are covered by the second background auxiliary theory used in this paper, described in more detail in Section 2.4.2.

We start by introducing the notation and terminology followed by an overview of related algebraic structures and related work. In Section 2, we present the three main results and conclude by the details on the structure and background theories of the formalization.

1.1 Notation and terminology

Words are finite sequences of elements from a given set Σ, where Σ is called an *alphabet*, and its elements are called *letters*. Accordingly, we represent words by the datatype of lists in our formalization, and the alphabet is typically represented in Isabelle by a type variable 'a. The set of all words over Σ is denoted by Σ^*, including the empty word, denoted by ε, which is represented as Nil or [] in Isabelle/HOL.

Words are endowed by the operation of *concatenation*, which corresponds to `append` for lists. Words with the operation of concatenation form a *free monoid*. The infix notation for the `append`-operation is @. For words, the concatenation is denoted by the multiplication sign $\cdot$ (which, as usual, is often omitted). We therefore allow, in our formalization, to write $\cdot$ instead of @. That is, $x \cdot y$ is equivalent to $x@y$. We write $u \leq_p v$ if u is a prefix of v, that is, if $v = u \cdot z$ for some z.

Seeing concatenation as a monoid multiplication naturally yields the concept of a power. We use the usual notation x^n of the n-th power of x in the mathematical text, and by $x^{@}n$ in the formalization. The set of all powers of a word t is usually denoted as t^* using the Kleene star familiar from regular expressions, where it is commonly used even for sets as, for example, in $\{u, v\}^*$. However, this allows a certain confusion. If G is a set of words over Σ, then G^* should denote all words over Σ generated by G. On the other hand, Σ^* denotes all words over the alphabet Σ, and the difference between the alphabet Σ and the set of words G has to be kept in mind. Strictly speaking, Σ^* is not generated by the alphabet

Σ, but rather by the set of singletons, that is, words of length one. While the difference between letters and singletons is typically ignored in the literature without any significant harm, the difference between a letter a, and the list $[a]$ must obviously be respected in the formalization. We therefore prefer to use the notation $\langle G\rangle$ for the submonoid of Σ^* generated by a set $G \subset \Sigma^*$. We also call it the *hull* of G. We nevertheless allow the expression $x \in t^*$ which is an abbreviation for $x \in \langle\{t\}\rangle$. The term `decompose` G u, abbreviated as $\mathrm{Dec}\, G\, u$, represents some decomposition of the word u into elements of G. It returns a list of words, i.e., of type 'a list list.

```
fun decompose :: 'a list set ⇒ 'a list ⇒ 'a list list (Dec - - ) where
  decompose G u = (SOME us. us ≠ ε ∧ us ∈ lists G ∧ u = concat us)
```

Hilbert's choice operator SOME is used here. The output of the function makes no good sense if the second argument is not in $\langle G\rangle$. Note, however, that even for elements of $\langle G\rangle$ the list is an unspecified choice among all possible decompositions. For example, if $G = \{a, ab, ba\}$ and $u = aba$, then $\mathrm{Dec}\, G\, u$ is either $[a, ba]$ or $[ba, a]$. This in particular implies that we cannot prove $\mathrm{Dec}\, G\, (u \cdot v) = \mathrm{Dec}\, G\, u \cdot \mathrm{Dec}\, G\, v$.

We deal with finite words only. An apparent exception is the infinite repetition $u^\omega = u \cdot u \cdot u \cdots$. However, this infinite word will be used exclusively in expressions of the form $w \leq_p u^\omega$, which is just a handy way of writing $\exists n. w \leq_p u^n$.

The length of a word w, that is the usual list `length`, is denoted by $|w|$. A word w of length n can be spelled as the list $[w_0, w_1, \ldots, w_{n-1}]$, where w_i represents the $(i+1)$-th letter of w. The first letter of a nonempty word w is also denoted hd w. The prefix of w of length $k \leq |w|$ is denoted $\mathrm{pref}_k w$ (`take` k w in Isabelle).

The word w has *a period* p if $1 \leq p$, and if $w_i = w_{i+p}$ for each $0 \leq i < |w| - p$. We allow (trivial) periods $p \geq |w|$.

One of our main interests is in *word equations*. Formally, a word equation is a pair of words $(L, R) \in X^* \times X^*$ over an alphabet X of *unknowns*. Nevertheless, the equation like $([x, y, z], [z, y, x])$ is usually written as $xyz = zyx$, a convention we already used above. A solution (in an alphabet Σ) of the equation (L, R) is a monoid morphism $f : X^* \to \Sigma^*$ (often called a *substitution*) such that $f(L) = f(R)$. (The defined concept should be more precisely described as *word equations without constants*. We do not deal with equations with constants in this paper.) The reader may further refer to Harju et al. [14].

1.2 Related algebraic structures and related work

Combinatorics of finite words focused on word equations has two basic aspects: the combinatorial and the algebraic. The combinatorial aspect is in an obvious way connected to words as lists, the algebraic aspect becomes important when considering a set of words as a generating set of a monoid. The algebraic aspect is exhibited and further discussed in Section 2.3 dedicated to the Graph Lemma. It is a basic decision of the formalization how to represent words in order to capture these two aspects. The first author in [21] conducted an inquiry into the possibility to deal with free monoids axiomatically. In particular, free monoids are fully characterized by the *equidivisibility property*:

```
lemma eqd: x · y = u · v ⟹ |x| ≤ |u| ⟹ ∃ t. x · t = u ∧ t · v = y
```

together with the provision that the length of possible decompositions of any element is bounded. Experience from this research confirms that the axiomatic approach has no

advantages. On the contrary, the elements of the free monoid will eventually be represented as lists of generators in any case (so in the Lean prover, for example, `free-monoid` over alphabet α is directly defined as a synonym for `list` α). Our formalization is therefore based on the datatype of lists. This fundamental datatype is well developed in Isabelle/HOL (as well as in all other provers), and we heavily build on the theory List.thy from the Main library, and the theory Sublist.thy from the HOL-Library.

Nevertheless, from the point of view of word equations, those theories contain only the solution of the easiest nontrivial word equation, namely $x \cdot y = y \cdot x$, showing that commuting words x and y are always powers of the same (shorter) word:

lemma comm-append-are-replicate:
xs @ ys = ys @ xs
$\Longrightarrow \exists$ m n zs. concat (replicate m zs) = xs $\wedge$ concat (replicate n zs) = ys

(We remark that this is the formulation in the 2021 release without redundant assumptions removed following our suggestion.) This result is called the *Commutation Lemma*. Since equations are our main interest, we improve readability using a slightly modified notation. Our version reads:

theorem comm-root: x $\cdot$ y = y $\cdot$ x $\longleftrightarrow$ ($\exists$ t. x $\in$ t* $\wedge$ y $\in$ t*)

Here t^* denotes the set $\{t^n \mid 0 \leq n\}$.

A similar remark concerning applicability for word equations applies to potentially related area of combinatorics of free groups, or even more generally, to combinatorial theory of (free) (semi)groups. The Isabelle/HOL theory Free-Groups by Breitner [6] contains fundamental properties of free groups including recently the Ping Pong lemma, which naturally exhibits some combinatorial features related to our work. Nevertheless, there is no direct overlap.

To our knowledge, the situation in other provers is not different. The most related to Combinatorics on Words is the Coq package Coq-Combi by Hivert [18] which uses specific parts of Combinatorics on Words results to prove some other results such as the Littlewood–Richardson rule. Another Coq package which is related is Coq-free-groups, formalizing elements of the free group theory (which is not as much developed as the above mentioned Isabelle/HOL free group theory by Breitner). Another related pieces of formalization can be found in the Lean Mathematical Library: it contains a basic formalization of free groups and free monoids, with no specific tools for submonoids of free groups (besides general submonoids).

Isabelle's *Archive of Formal Proofs* [1] contains a large group of theories on Automata and formal languages. The Coq package Coq-automata is situated within the same topic. However, there is almost no overlap with word equations and questions we are interested in. For example, the theory of regular expressions (or, more generally, Kleene algebras) deals with structures on sets of languages, not with individual languages, which moreover typically are not themselves monoids. We can illustrate this by one of our recent formalizations [20]. It is a basic property of regular languages to be closed under intersection. However, to classify possible intersections $\{x, y\}^* \cap \{u, v\}^*$ of two monoids generated by pairs of non-commuting words is a nontrivial task, which has little to do with finite automata or with a general theory of regular languages.

It should be stressed that monoids as such are too general a structure, and do not provide any significant theoretical support for reasoning about lists. The main defining property of

monoids, associativity, is captured by lists trivially. The single exception are properties of powers. We therefore interpret lists as an instance of the class `monoid-mult`:

interpretation monoid-mult ε append

This immediately yields a series of claims like

lemma power-add-list: $x^{@}n \cdot x^{@}m = x^{@}(n+m)$

where $x^{@}n$ is our notation for the interpreted `power`.

2 Presented results

2.1 The Periodicity Lemma

Periodicity is one of the most important and most studied properties of words. In our formalization, we use the following definition:

definition periodN :: 'a list ⇒ nat ⇒ bool
where periodN w n = w ≤p (take n w)$^\omega$

A related definition is the definition of the *period root*:

definition period-root :: 'a list ⇒ 'a list ⇒ bool (- ≤p -$^\omega$)
where [simp]: period-root u r = (u ≤p r · u ∧ r ≠ ε)

with the notation $u \leq_p r^\omega$. This notation is justified by the observation that the following claims are equivalent:

- w has a period p (in the sense given in Section 1.1);
- w is a prefix of $u \cdot w$, where u is a word of length p (the period root);
- w is a prefix of u^ω.

A word can have more than one period. This possibility is regulated by the following famous result.

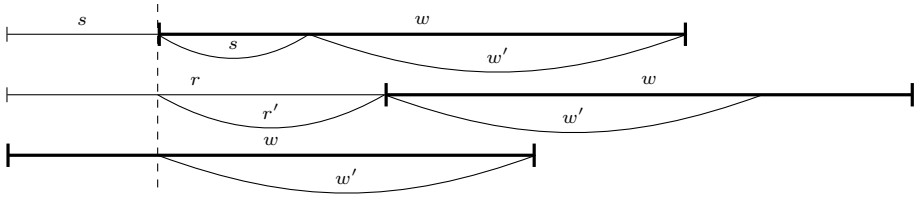
▶ **Lemma 1** (Periodicity Lemma [10]). *If a word w of length at least $p + q - \gcd(p, q)$ has periods p and q, then it also has a period $\gcd(p, q)$.*

The proof is a combination of two elementary facts. The first one is the above mentioned characterization of the period by the period root. The second one is the Commutation Lemma. We first prove the following claim, which can be seen as a modification of the Euclidean algorithm.

▶ **Lemma 2.** *Let $w \leq_p r \cdot w$ and $w \leq_p s \cdot w$. If $|r| + |s| - \gcd(|s|, |r|) \leq |w|$, then $r \cdot s = s \cdot r$.*

Proof. The assumptions imply that both s and r are prefixes of w. By symmetry, we can suppose $|s| \leq |r|$ which yields $s \leq_p r$. Let r' and w' be such that $r = s \cdot r'$ and $w = s \cdot w'$.

Then s, r' and w' satisfy the assumptions of the claim, see the following figure.



In particular, we have

$$|r'| + |s| - \gcd(|s|, |r'|) = |r| - |s| + |s| - \gcd(|s|, |r| - |s|) =$$
$$|r| + |s| - \gcd(|s|, |r|) - |s| \leq |w| - |s| = |w'|.$$

If $s = \varepsilon$, the claim holds. If s is nonempty, then we have that s and r' commute by induction on $|s| + |r|$. Hence also s and r commute. ◀

The proof of the Periodicity lemma is now easily concluded using the Commutation Lemma (see Section 1.2):

Proof of the Periodicity lemma. Assume $p \leq q$, and let t be the common root of $s = \mathrm{pref}_p w$ and $r = \mathrm{pref}_q w$. Then $|t|$ divides $\gcd(p, q)$. Since w is a prefix of s^ω, it is also a prefix of t^ω, hence it has a period $\gcd(p, q)$. ◀

We want to point out, based on this very simple example, several observations. First, we note the interplay between intuition brought about by the picture in the above proof, and the formal manipulation. In order to make the induction step, namely to see that both $w' \leq_p r' \cdot w$ and $w' \leq_p s \cdot w'$, one can either consult the picture, or use a formal verification which consists in the following considerations:

1. cancellation of s from $w \leq_p s \cdot w$ after substitution of both occurrences of w with $s \cdot w'$ yields $w' \leq_p s \cdot w'$;
2. cancellation of s from $w \leq_p s \cdot w$ after substitution of just the first occurrence of w with $s \cdot w'$ yields $w' \leq_p w$;
3. cancellation of s from $w \leq_p r \cdot w$ yields $w' \leq_p r' \cdot w$;
4. the latter and $w' \leq_p w$ yields $w' \leq r' \cdot w'$.

Actually, the last step still requires a simple length argument.

Although a similar point could be probably made about mathematical proofs in general, in Combinatorics on words, thanks to the elementary character of lists, the gap between the insight and the formal proof is very typical. Calibrating the right mixture of the insight and the detail, which is naturally very reader-specific, is an almost impossible task. One of the main advantages of the formalization becomes apparent here: it allows to focus on ideas while being sure that no unexpected gaps were missed.

Another lesson from this example is that it was in the context of this formalization that we realized how important and useful the equivalent characterization of periods are. More precisely, the formalization makes clear that the characterization by the period root is "the right one". In fact, we believe that the proof presented here is the shortest one available in the literature. The Periodicity lemma has many different proofs, several of them presented already in the original paper by Fine and Wilf [10]. Proofs based on the numeric definition of period by indexes (that is, by $w_i = w_{i+p}$) can be rather involved (see, for example, the basic reference monograph [26]). Our proof is close to the version in Berstel an Karhumäki [3] but without the need to deal separately with the case when the periods are not coprime.

The superiority of the periodic root definition of a period can be captured as its suitability for equational reasoning. We add another example of this phenomenon. Consider the following claim:

▶ **Lemma 3.** *If $x \cdot y = z$ and the words x and z commute, then also y and z commute.*

This is a trivial claim which would be justified (if needed) as follows:

Proof. Commuting words are powers of the same word. Canceling x from $x \cdot y = z$ therefore yields that also y is the power of the same word. ◀

This appeal to the Commutation Lemma is an almost instinctive move for a researcher in Combinatorics on Words. However, this argument does not seem to be sufficiently trivial for an automated tool (like `try0` in Isabelle). Nevertheless, the proof is the simple **by** `force` anyway, since Isabelle employs a different approach, which is humanly less transparent but is based on a simple manipulation of equalities.

Proof. Substitute $x \cdot y$ for z in $x \cdot z = z \cdot x$ to obtain $x \cdot x \cdot y = x \cdot y \cdot x$. Cancel x and multiply both sides by y from right to obtain $x \cdot y \cdot y = y \cdot x \cdot y$, which is the desired equality after substituting z back for $x \cdot y$. ◀

Finally, a particular challenge for the formalization of the Periodicity Lemma is the humanly obvious argument from symmetry (cf. Harrison [17]), which allows to assume that s is not longer than r. This move is sometimes dealt with in formalization by defining s_1 and r_1 as the shorter and the longer of the two words respectively, and then carrying out the proof using s_1 and r_1. This approach is nevertheless quite tedious, in particular in proofs by induction. We use a little trick to deal with this problem: the induction is made not simply on $|s| + |r|$ but rather on $|s| + |s| + |r|$. Then, considering the cases $|r| < |s|$ and $|s| \leq |r|$, the former case is covered by the induction hypothesis exactly by symmetry of s and r as in the informal proof.

The bound in the Periodicity Lemma is optimal in the following sense:

▶ **Lemma 4.** *Let p and q be positive integers such that $p \nmid q$ and $q \nmid p$. Then there is a word of length $p + q - \gcd(p, q) - 1$ that has periods p and q, and not a period $\gcd(p, q)$.*

The word from the lemma is called an FW-word(p, q) (for Fine and Wilf). With the additional requirement that it contains maximum number of distinct letters, it is unique up to renaming of letters (this property is not proved in our formalization). Such a word FW-word(p, q), which is equal to FW-word(q, p), with the maximum number of distinct letters can be constructed as follows. Use natural numbers as the alphabet, and let $[n]$ denote the word $\mathtt{0} \cdot \mathtt{1} \cdots (\mathtt{n} - \mathtt{1})$. Assume $p < q$ and let $d = \gcd(p, q)$. If $p = kd$ and $q = (k + 1)d$, $1 < \mathrm{k}$, then the word

$$\text{FW-word}(p, q) = [d]^{k-1} \cdot [d - 1] \cdot d \cdot [d]^{k-1} \cdot [d - 1]$$

satisfies the required conditions. Otherwise FW-word(p, q) is defined inductively as the prefix of $(\text{FW-word}(p, q - p))^\omega$ of the required length. The correctness of the construction can be proved as follows:

Proof. If $q = p + d$, then the word FW-word(p, q) defined above has the required properties as can be directly verified. If $q = p + kd$ with $1 < k$, then kd does not divide p and by induction we obtain a word v of length $q - d - 1 = (q - p) + p - d - 1 > \max(p, q - p)$,

which has periods p and $q-p$ and does not have a period d. The word v is then a prefix of $(\mathrm{pref}_p v)^\omega$ and of $(\mathrm{pref}_{q-p} v)^\omega$. It is therefore also a prefix of words $\mathrm{pref}_p v \cdot v$ and $\mathrm{pref}_{q-p} v \cdot v$. Consider the prefix w of $(\mathrm{pref}_p v)^\omega$ of length $p+q-d-1 > q$. The word w has a period p since it is a prefix of $(\mathrm{pref}_p v)^\omega$, and it does not have the period d since v is a prefix of w. It remains to show that w has a period q, that is, that w is a prefix of $\mathrm{pref}_q w \cdot w$. First, note that $w = \mathrm{pref}_p v \cdot v$, hence $\mathrm{pref}_q w = \mathrm{pref}_p v \cdot \mathrm{pref}_{q-p} v$. Since v is a prefix of $\mathrm{pref}_{q-p} v \cdot v$, we have that w is a prefix of $\mathrm{pref}_p v \cdot \mathrm{pref}_{q-p} v \cdot \mathrm{pref}_p v = \mathrm{pref}_q w \cdot \mathrm{pref}_p v$, which is a prefix of $\mathrm{pref}_q w \cdot w$. ◀

We have implemented the above construction, and formalized the proof of its correctness:

```
theorem fw-word: assumes ¬ p dvd q and  ¬ q dvd p
  shows |FW-word p q| = p + q − gcd p q − 1 and
        periodN (FW-word p q) p and
        periodN (FW-word p q) q and
        ¬ periodN (FW-word p q) (gcd p q)
```

The formalized proof is relatively long (over 200 lines). This reflects the number of facts that have to be verified, including the shifty claim about the "direct verification" of the base case which spans more than half of the proof.

We thereby provide a formally verified calculation of an FW-word. Here are some sample values:

```
value FW-word 3 7
```

[0, 0, 1, 0, 0, 1, 0, 0]

```
value FW-word 4 6
```

[0, 1, 0, 2, 0, 1, 0]

```
value FW-word 12 18
```

[0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4, 6, 0, 1, 2, 3, 4, 5, 0, 1, 2, 3, 4]

2.2 The theorem of Lyndon and Schützenberger

The very first folklore result in the basic course of Combinatorics of Words is the Commutation Lemma mentioned above, solving the equation $x \cdot y = y \cdot x$. The Commutation Lemma is easy to prove directly, but it can be also noted that the word $w = uv = vu$ has periods $|u|$ and $|v|$, and the claim follows from the Periodicity Lemma.

Moved from two to three unknowns, solving equations becomes a challenging task. Although a classification of monoids generated by three words is available (see a survey by Harju and Nowotka [15]), it is a complex one. Recall that the question about the maximal number of independent equations in three unknowns remains open as mentioned in the Introduction. From this point of view, the LSE, i.e. the equation $x^a \cdot y^b = z^c$ with $2 \leq a, b, c$ solved by Lyndon and Schützenberger in 1962, is important both historically and conceptually. As already mentioned, this equation with three unknowns was originally solved in a more general case of a free group, but it has been subsequently further investigated in free monoids, and several alternative proofs have been suggested for example by Dömösi and Horváth [8] or Harju and Nowotka [16]. It would be interesting to formalize the original proof in free groups, however this task goes beyond our present focus. We expect that the proof in free

groups could not be simplified as the word variant we present below. Note that the equation can be seen as a natural follow up of the Periodicity Lemma since it deals with a special configuration of three distinct periods.

theorem Lyndon-Schutzenberger:
assumes x[@]a·y[@]b = z[@]c **and** 2 ≤ a **and** 2 ≤ b **and** 2 ≤ c
shows x·y = y·x **and** x·z = z·x **and** y·z = z·y

We present here a concise formalization of the theorem of Lyndon and Schützenberger in free monoids. We first give a full paper proof that we formalized. It is similar to the one given in [26, Section 9.2], however, the core case $c = 3$ is significantly simplified.

Proof. By symmetry, assume $|x^a| \geq |y^b|$.

The word x^a has periods $|x|$ and $|z|$. If $|x^a| \geq |z| + |x|$, then the Periodicity Lemma implies that x and z have a period dividing $|x|$ and $|z|$, which easily yields that they commute. Similarly if $|y^b| \geq |z| + |y|$.

Therefore, suppose that x^{n-1} is a proper prefix of z and y^{m-1} a proper suffix of z. Then $|x^a| < 2|z|$ and $|y^b| < 2|z|$, hence $c < 4$.

Let $c = 3$. If $a \geq 3$, then $|x^2| < |z|$ implies $|x^3| < \frac{3}{2}|z|$, contradicting the assumption $|x^a| \geq |y^b|$. Therefore $a = 2$ and $|x| \geq |y|$. There are words u, v, w such that $x = uw = wv$, $z = xu = wvu$ and $y^b = vuwvu$. From $uw = wv$ we deduce that uwv has a period $|u|$. Moreover, uwv is a factor of y^b which implies that it has a period $|y|$. Since $|y| + |u| \leq |uwv|$, the Periodicity Lemma implies that $d = \gcd(|u|, |y|)$ is a period of uwv. It is easy to see that d divides also $|v|$ and $|w|$, which implies that words u, v and w commute. Therefore also x, y and z commute.

The case $c = 2$ remains. We have $z = x^{a-1}u = wy^b$, where $uw = x$. Then $wz = (wu)^a = w^2y^b$, where wu is shorter than z. By induction on $|z|$, we obtain that w, y and wu commute. Therefore also x, y and z commute. ◀

In the formalization, we first prove that x and y commute:

lemma per-lemma-case:
assumes |z| + |x| ≤ |x[@]a|
shows x·y=y·x

The other two commutation claims, humanly obvious consequences of the first one, are proved relatively easily using auxiliary lemmas about roots formalized in our background theory.

Two of the three cases in the proof are proven as separate lemmas. Namely, the case solved by the Periodicity Lemma:

lemma per-lemma-case:
assumes |z| + |x| ≤ |x[@]a| **and** x ≠ ε
shows x·y=y·x

and the core case $c = 3$:

lemma core-case:
assumes
c = 3 **and**
b∗|y| ≤ a∗|x| **and** x ≠ ε **and** y ≠ ε **and**
lenx: a∗|x| < |z| + |x| **and**
leny: b∗|y| < |z| + |y|
shows x·y = y·x

It would seem natural to solve even the remaining case $c = 2$ separately, and then simply put the three cases together. However, this is not possible, since the induction, abruptly announced at the end of the paper proof, actually governs the whole proof since it covers the first two cases as well. (This is one of the typical backtracking moments of the development.) We conclude this section noting that also in this case we use a similar trick to deal with the symmetry as in the proof of the Periodicity lemma. Namely, the the induction is on $|z| + b|y|$. If $|x^a| < |y^b|$, then we switch to the symmetric case which yields the result immediately by induction.

2.3 The Graph Lemma

In order to present the third classical result, the Graph Lemma, we first need to explain its algebraic background which is covered by our second auxiliary formalized theory. It is immediate that (unlike in the free group case) submonoids of the free monoid are not always free. Consider, for example, the monoid $M = \langle\{aa, aab, baa\}\rangle$ generated by words aa, aab and baa. While $\{aa, aab, baa\}$ is its *basis*, denoted $\mathfrak{B}M$,that is, the minimal generating subset (which is unique for submonoids of the free monoid), the monoid M is not free since $aab \cdot aa = aa \cdot baa$ are two distinct decompositions of the word $aabaa$ into elements of the basis. In other words, $x \mapsto aa$, $y \mapsto baa$, $z \mapsto aab$ is a solution of the equation $x \cdot y = z \cdot x$. On the other hand, each set G of words has a *free hull* $\langle G\rangle_F$, the unique smallest free monoid containing G. This can be seen using another equivalent characterization of free monoids, namely the *stability condition*:

$$p, pw, wq, q \in M \implies w \in M. \tag{1}$$

We remark that the equality $p \cdot wq = pw \cdot q$ provides a link to the equidivisibility property, another equivalent characterization of freeness mentioned in Section 1.2. Since the stability condition is obviously closed under intersection, we obtain

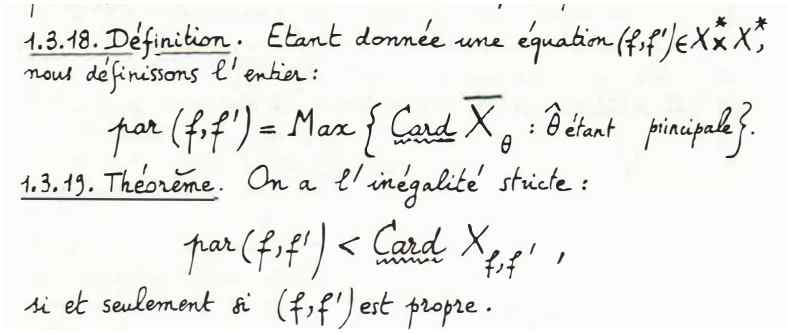
▶ **Lemma 5.**

$$\langle G\rangle_F = \bigcap\{M \mid G \subseteq M, M \text{ is free}\}.$$

For example, the free hull of $G = \{aa, aab, baa\}$ is $\langle\{aa, b\}\rangle$. The basis of $\langle G\rangle_F$ is also called the *free basis* of G, and is denoted $\mathfrak{B}_F\ G$. The key (and defining) property of *free* monoids is uniqueness of the decomposition into elements of the basis. That is, Dec $(\mathfrak{B}_F\ G)$ is a well defined decomposition function for any G. In our example, we have Dec $(\mathfrak{B}_F\ G)\, aabaa = [aa, b, aa]$. If some set G is equal to its free basis, that is, if it is the minimal generating set of a free monoid, then G is called a *code*.

If $f : X^* \to \Sigma^*$ is a morphism (a solution of a word equation), then its *rank* is the cardinality of the free basis of the set of images $\{f(x) \mid x \in X\}$. The fact that any solution of a nontrivial equation has rank less than the number of unknowns is sometimes called "a

defect effect". It was probably for the first time proved in the book by Lentin [25] which curiously exists in the hand-written form only:


1.3.18. Définition. Etant donnée une équation $(f,f') \in X^* \times X^*$, nous définissons l'entier :

$$\text{par}(f,f') = \text{Max}\left\{ \text{Card}\, \overline{X}_\theta : \hat{\theta} \text{ étant principale} \right\}.$$

1.3.19. Théorème. On a l'inégalité stricte :

$$\text{par}(f,f') < \text{Card}\, X_{f,f'},$$

si et seulement si (f,f') est propre.

However, unlike the case of linear equations mentioned in the Introduction, word equations do not allow a straightforward cumulative defect effect. In other words, there can be large systems of independent word equations (see Karhumäki and Plandowski [24]).

The Graph Lemma is a result enforcing a weak but very general form of the cumulative defect effect. It owes its name to the formulation by Harju and Karhumäki [13]. We illustrated the graph in question by an example in the Introduction. The proof of the Graph Lemma that we formalize here is from Berstel et al. [4]. The claim in this formulation reads as follows:

▶ **Theorem 6** (Graph Lemma). *Let G be a set of words. Then*

$$\mathfrak{B}_F\ G = \{\mathrm{hd}(\mathrm{Dec}\,(\mathfrak{B}_F\ G)\,x) \mid x \in G, x \neq \varepsilon\}\,.$$

This is related to the graph described in the Introduction in the following way. The theorem says that each element of the basis appears as the head in the decomposition of some $x \in G$. Consider again the system of equations

$$xyz = yzx$$
$$xzy = zyx$$

and let f be its solution. From $f(xyz) = f(yzx)$ we deduce that $\mathrm{hd}(f(x)) = \mathrm{hd}(f(y))$. Similarly, we have $\mathrm{hd}(f(x)) = \mathrm{hd}(f(z))$ from $f(xzy) = f(zyx)$. The Graph Lemma now implies that the rank of f is one, yielding a cumulative defect effect: each equation decreased the rank of the solution by one.

The proof of the Graph Lemma has two steps. We first prove the following lemma:

▶ **Lemma 7.** *Let C be a code and let $b \in C$. Then also*

$$C' = \{zb^k \mid z \in C, z \neq b\}$$

is a code, and it generates the submonoid $S = \{x \in \langle C\rangle \mid \mathrm{hd}\, z \neq b\}$ of $\langle C\rangle$.

This lemma is considered to be humanly obvious. In [4] (see p. 171), this is not even formulated as a separate lemma, and the claim is justified by a simple appeal to intuition: any word not starting with b has a unique decomposition into elements of C'. On the other hand, the formalization of this claim is challenging. Indeed, the lemma actually contains (at least) the following claims:

- C' is a basis;
- C' generates S;

- C' is a code,

each of which requires nontrivial formalization effort.

Having proved Lemma 7, we can prove the Graph Lemma by contradiction. If $b \in \mathfrak{B}_F\ G$ is not a head of any decomposition, then G is contained in $\langle C' \rangle$ where

$$C' = \{zb^k \mid k \geq 0,\ z \in \mathfrak{B}_F\ X,\ z \neq b\}$$

is a code. Since $\langle C' \rangle$ does not contain b, we have $\langle C' \rangle \subsetneq \langle G \rangle_F$, a contradiction with Lemma 5.

2.4 Overview of the structure of the published formalization

The formalization is published in the Gitlab repository [36] as a part of an evolving Combinatorics on Words formalization project. The content described in this article is covered by the following five theories:

- **Basics/CoWBasic.thy**: defines basic concepts, and contains more than five hundred auxiliary lemmas (not all of them needed for the three main presented results);
- **Basics/Submonoids.thy**: defines submonoids, and contains the algebraic backbone: submonoids, fundamental properties of bases, codes and free hulls;

and three more advanced and more specific theories:

- **Basics/Periodicity_Lemma.thy**: contains the periodicity lemma, along with the proof of its optimality;
- **Basics/Lyndon_Schutzenberger.thy**: covers the Lyndon-Schützenberger theorem;
- **Graph_Lemma/Graph_Lemma.thy**: contains the Graph Lemma and its application to binary codes.

We describe the two background theories, CoWBasic and Submonoids, in more detail in the next two sections.

2.4.1 CoWBasic background theory

As already mentioned, the theory CoWBasic serves as a basis for a formalization of a Combinatorics on Words results such as the three results presented in this article. Its purpose is to cover elementary concepts (the "folklore" mentioned in Introduction) using a common notation and theorem formulation, and thus make them ready to be used by a Combinatorics on Words researcher.

CoWBasic is builds heavily on the Main's theory List and on the theory HOL-Library.Sublist. Besides the definition of the fundamental datatype list, the first mentioned theory contains many Combinatorics on Words relevant concepts such as the functions `take`, `drop`, `rotate`, `concat`, and `length`, accompanied by many relevant lemmas. The theory HOL-Library.Sublist extends the range of available tools by defining `prefix`, `longest-common-prefix`, `suffix`, and (contiguous) `sublist`, again furnished with many relevant claims. As summarized in Section 1.1, the theory first establishes some elementary prevalent notation in Combinatorics on Words. It extends the coverage of supporting claims related existing concepts ranging from observation level lemmas such as

> **lemma** pref-drop: u ≤p v ⟹ drop p u ≤p drop p v

to slightly more elaborate (in terms of a formal proof) claims such as

> **lemma** rotate-back: **obtains** m **where** rotate m (rotate n u) = u.

Most of the claims themselves can be considered quite simple, i.e., a human reader, not necessarily an expert in Combinatorics on Words, would consider them "obvious" or maybe requiring a simple argument or a picture (cf. the discussion in Section 2.1). Naturally, many of these lemmas are implicitly used in paper proofs hidden under claims such as "It easily follows". The selection of these auxiliary claims is based first on our consideration, second on the actual need in the formalization of more advanced results. As the development is an iterative process, many definitions and lemmas are results of several optimizations based on our usage experience.

In the same spirit, the theory CoWBasic introduces new concepts and supporting claims. While some of these were mentioned along with the main presented results in Section 2, we list here some most prominent other examples. We define the left quotient of a word as follows:

definition left-quotient:: 'a list ⇒ 'a list ⇒ 'a list ((-$^{-1>}$)(-))
where left-quotient-def[simp]: left-quotient u v = (THE z. u · z = v).

A word is primitive if it is not a power of some other word:

definition primitive :: 'a list ⇒ bool
where primitive u = (∀ r k. r$^{@}$k = u ⟶ k = 1)

Given a non-empty word w which is not primitive, it is natural to look for the shortest u such that $w = u^k$. Such a word is primitive, and it is the primitive root of w:

definition primitive-root :: 'a list ⇒ 'a list ⇒ bool (- $\in_p$ - ∗)
where primitive-root x r = (x ≠ ε ∧ x ∈ r* ∧ primitive r)

2.4.2 Submonoids background theory

Whereas the first auxiliary theory overlaps with existing tools, Submonoids theory develops its own tools, building on CoWBasic. Its main purpose is to cover algebraic properties of submonoids of a free monoids, a background needed for the Graph Lemma and already introduced in Section 2.3.

The first two notions were already introduced in Section 1.1, the first is the *hull*:

inductive-set hull :: 'a list set ⇒ 'a list set (⟨-⟩)
for G **where**
emp-in: ε ∈ ⟨G⟩
|prod-cl: w1 ∈ G ⟹ w2 ∈ ⟨G⟩ ⟹ w1 · w2 ∈ ⟨G⟩

and the second is *a* decomposition of a word into some sequence of words, i.e., the function `decompose` (abbreviated as Dec).

The remaining notions introduced in Section 2.3 follow. It is a noteworthy fact that their definitions are slightly different from the "paper" version above. This difference is motivated purely by a more suitable use in the formalization, based on authors' experience with primordial versions of the formalization using exactly the "paper" versions. Basis relies on the notion of a *simple element*:

```
function simple-element :: 'a list ⇒ 'a list set ⇒ bool ( - ∈B - ) where
  simple-element b G = (b ∈ G ∧ (∀ us. us ≠ ε ∧ us ∈ lists G ∧ concat us = b ⟶ |us|
= 1))
```

Basis is then the set of all simple elements:

```
fun basis :: 'a list set ⇒ 'a list set (𝔅 - ) where
  basisdef: basis G = {x. x ∈B G}
```

The definition stated above is shown as a pair of theorems – the basis is the minimal generating set:

```
theorem ⟨𝔅 G⟩ = ⟨G⟩
theorem ⟨S⟩ = ⟨G⟩ ⟹ 𝔅 G ⊆ S
```

The concept of a *code*, implemented as a locale, is formalized as

```
locale code =
  fixes 𝒞
  assumes 𝒞-is-code: xs ∈ lists 𝒞 ⟹ ys ∈ lists 𝒞 ⟹ concat xs = concat ys ⟹ xs = ys
```

and finally the inductive definition of the *free hull* reads

```
inductive-set free-hull :: 'a list set ⇒ 'a list set (⟨-⟩F)
  for G where
    ε ∈ ⟨G⟩F
  | free-gen-in: w ∈ G ⟹ w ∈ ⟨G⟩F
  | w1 ∈ ⟨G⟩F ⟹ w2 ∈ ⟨G⟩F ⟹ w1 · w2 ∈ ⟨G⟩F
  | p ∈ ⟨G⟩F ⟹ q ∈ ⟨G⟩F ⟹ p · w ∈ ⟨G⟩F ⟹ w · q ∈ ⟨G⟩F ⟹ w ∈ ⟨G⟩F
```

The freeness is ensured by the last condition which is the stability condition (1). The fact that the free hull is the smallest free monoid containing the generating set is again proven as a theorem:

```
theorem free-hull-inter: ⟨G⟩F = ⋂ {M. G ⊆ M ∧ M = ⟨M⟩F}
```

Finally, free basis is exactly as introduced above, namely $\mathfrak{B}_F\ G = \mathfrak{B}\ \langle G\rangle_F$:

```
definition free-basis :: 'a list set ⇒ 'a list set (𝔅F -)
  where free-basis G ≡ 𝔅 ⟨G⟩F
```

3 Conclusion

The aim of this paper is to introduce an ongoing formalization of Combinatorics on Words. The next step after the Lyndon-Schützenberger theorem is its natural extension obtained independently by J.-P. Spehner [33], and by E. Barbin-Le Rest, M. Le Rest [2] which claims that $x^i y$ is the only non-trivial way (up to symmetry and conjugation) how two

non-commuting words can form a non-primitive word (like z^c). The history of this result is another good motivation for our formalization project. The result, while very natural and important, has been almost forgotten (it was cited only six times before 2015). A weaker form of this result was even rediscovered in 1994 [32], and started to be referenced. One reason for this is that already this relatively simple result is very technical and difficult to read. Moreover, the paper contains several minor inaccuracies which makes the reading even more labored. This is by no means an exceptional situation in Combinatorics on Words, which testifies for a strong need of formally verified proofs in the field.

References

1 Archive of Formal Proofs. https://www.isa-afp.org/topics.html.

2 Evelyne Barbin-Le Rest and Michel Le Rest. Sur la combinatoire des codes à deux mots. *Theor. Comput. Sci.*, 41:61–80, 1985. doi:10.1016/0304-3975(85)90060-X.

3 J Berstel and J Kkarhumäki. Combinatorics on Words – a tutorial. In *Current Trends in Theoretical Computer Science*, pages 415–475. World Scientific, April 2004. doi:10.1142/9789812562494_0059.

4 J Berstel, D Perrin, J.F Perrot, and A Restivo. Sur le théorème du défaut. *Journal of Algebra*, 60(1):169–180, 1979. doi:10.1016/0021-8693(79)90113-3.

5 Jean Berstel and Dominique Perrin. The origins of combinatorics on words. *European Journal of Combinatorics*, 28(3):996–1022, 2007. doi:10.1016/j.ejc.2005.07.019.

6 Joachim Breitner. Free groups. *Archive of Formal Proofs*, 2010. , Formal proof development. URL: https://isa-afp.org/entries/Free-Groups.html.

7 Volker Diekert. Makanin's algorithm. In *Algebraic Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, pages 387—442. Cambridge University Press, 2002. doi:10.1017/CBO9781107326019.013.

8 Pál Dömösi and Géza Horváth. Alternative proof of the Lyndon–Schützenberger theorem. *Theoretical Computer Science*, 366(3):194–198, 2006. Automata and Formal Languages. doi:10.1016/j.tcs.2006.08.023.

9 Georges Gonthier et al. A machine-checked proof of the odd order theorem. In *ITP*, volume 7998 of *Lecture Notes in Computer Science*, pages 163–179. Springer, 2013.

10 N. J. Fine and H. S. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16(1):109–109, January 1965. doi:10.1090/S0002-9939-1965-0174934-9.

11 Georges Gonthier. Formal proof—the four-color theorem. *Notices Amer. Math. Soc.*, 55(11):1382–1393, 2008.

12 Thomas Hales et al. A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi*, 5:e2, 2017. doi:10.1017/fmp.2017.1.

13 T. Harju and J. Karhumäki. On the defect theorem and simplifiability. *Semigroup Forum*, 33:199–217, 1986.

14 Tero Harju, Juhani Karhumäki, and Wojciech Plandowski. Independent systems of equations. In *Algebraic Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, pages 443—471. Cambridge University Press, 2002. doi:10.1017/CBO9781107326019.014.

15 Tero Harju and Dirk Nowotka. On the independence of equations in three variables. *Theoretical Computer Science*, 307(1):139–172, 2003. WORDS. doi:10.1016/S0304-3975(03)00098-7.

16 Tero Harju and Dirk Nowotka. The equation $x^i = y^j z^k$ in a free semigroup. *Semigroup Forum*, 68(3):488–490, 2004. doi:10.1007/s00233-003-0028-6.

17 John Harrison. Without loss of generality. In Stefan Berghofer, Tobias Nipkow, Christian Urban, and Makarius Wenzel, editors, *Theorem Proving in Higher Order Logics*, pages 43–59, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

18 Florent Hivert et al. Coq-Combi. https://github.com/hivert/Coq-Combi, 2021.

19 Štěpán Holub. Commutation and beyond. In Srečko Brlek, Francesco Dolce, Christophe Reutenauer, and Élise Vandomme, editors, *Combinatorics on Words*, pages 1–5, Cham, 2017. Springer International Publishing.

20 Štěpán Holub and Štěpán Starosta. Binary intersection formalized. *Theor. Comput. Sci.*, to appear.

21 Štěpán Holub and Robert Veroff. Formalizing a fragment of combinatorics on words. In Jarkko Kari, Florin Manea, and Ion Petre, editors, *Unveiling Dynamics and Complexity*, pages 24–31, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-58741-7_3.

22 Artur Jez. Recompression: A simple and powerful technique for word equations. *J. ACM*, 63(1):4:1–4:51, 2016. doi:10.1145/2743014.

23 Artur Jeż. Word equations in nondeterministic linear space. In *44th International Colloquium on Automata, Languages, and Programming*, volume 80 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 95, 13. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017.

24 Juhani Karhumäki and Wojciech Plandowski. On the size of independent systems of equations in semigroups. *Theoretical Computer Science*, 168(1):105–119, 1996. doi:10.1016/S0304-3975(96)00064-3.

25 A. Lentin. *Equations dans les monoides libres*. De Gruyter Mouton, 1972. doi:10.1515/9783111544526.

26 M. Lothaire. *Combinatorics on words*. Cambridge Mathematical Library. Cambridge University Press, Cambridge, 1997. doi:10.1017/CBO9780511566097.

27 R. C. Lyndon and M. P. Schützenberger. The equation $a^m = b^n c^p$ in a free group. *Michigan Math. J.*, 9(4):289–298, December 1962. doi:10.1307/mmj/1028998766.

28 Gennadiy Semenovich Makanin. The problem of solvability of equations in a free semigroup. *Matematicheskii Sbornik*, 145(2):147–236, 1977.

29 Dirk Nowotka and Aleksi Saarela. One-variable word equations and three-variable constant-free word equations. *Int. J. Found. Comput. Sci.*, 29(5):935–950, 2018. doi:10.1142/S0129054118420121.

30 Dirk Nowotka and Aleksi Saarela. An optimal bound on the solution sets of one-variable word equations and its consequences. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 136, 13. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2018.

31 Aleksi Saarela. Independent systems of word equations: From Ehrenfeucht to eighteen. In Robert Mercaş and Daniel Reidenbach, editors, *Combinatorics on Words*, pages 60–67, Cham, 2019. Springer International Publishing.

32 H.J. Shyr and S.S. Yu. Non-primitive words in the language p^+q^+. *Soochow Journal of Mathematics*, 20, January 1994.

33 J.-P. Spehner. *Quelques problèmes d'extension, de conjugaison et de presentation des sous-monoïdes d'un monoïde libre.* PhD thesis, Université Paris VII, Paris, 1976.

34 Axel Thue. Über unendliche Zeichenreichen. *Skrifter: Matematisk-Naturvidenskapelig Klasse*, 1906.

35 Axel Thue. Uber die gegenseitige lage gleicher teile gewisser zeichenreihen. *Kra. Vidensk. Selsk. Skrifer, I. Mat. Nat. Kl.*, pages 1–67, 1912.

36 Štěpán Holub, Štěpán Starosta, et al. Combinatorics on words formalized (release v1.3). https://gitlab.com/formalcow/combinatorics-on-words-formalized, 2021.